

Peano Commutativity in [S']

Martin Røpcke

1. juli 2005

Indhold

1	Introduction	3
2	The Logiweb System	3
2.1	References	4
2.2	The Base Page	4
3	A Logiweb Page	5
3.1	“Main Menu”	5
3.2	The Pyk Language	5
3.3	The Structure of a Logiweb Page	6
4	Axiomatic Theories	8
4.1	The Axiomatic Theory L	9
4.2	An Axiomatic Logiweb Theory	9
5	Peano Arithmetic	11
5.1	Peano Arithmetic in Logiweb	11
5.1.1	S' Axioms	12
5.1.2	S' Rules of Inference	12
5.1.3	S' Theorems	12
6	Proof of Commutativity of Addition	13
6.1	M Proposition 3.2(a)	13
6.2	M Tautology A	14
6.3	M Proposition 3.2(b)	14
6.4	M Tautology B	15
6.5	M Proposition 3.2(c)	15
6.6	M Proposition 3.2(d)	16
6.7	M Proposition 3.2(f)	16
6.7.1	M Proposition 3.2(f) (i)	16
6.7.2	M Proposition 3.2(f) (ii)	17
6.7.3	M Proposition 3.2(f)	18
6.8	M Proposition 3.2(g)	18

6.8.1	M Proposition 3.2(g) (i)	18
6.8.2	M Proposition 3.2(g) (ii)	19
6.8.3	M Proposition 3.2(g)	20
6.9	M Proposition 3.2(h)	20
6.9.1	M Proposition 3.2(h)(i)	20
6.9.2	M Proposition 3.2(h) _(g) (ii)	21
6.9.3	M Proposition 3.2(h)(ii)	22
6.9.4	M Proposition 3.2(h)	23
7	Conclusion	23
A	The Name of the Page	25
B	Priority Table	25
C	Bibliography	28

1 Introduction

This report is turned in for credit for the course “Logic” at DIKU summer 2005. In short, the assignment is to prove the commutativity of addition within peano arithmetic using the Logiweb system developed by Klaus Grue. Logiweb will be introduced shortly. That is, in this report we shall carry out a proof of the proposition that for all natural number the following holds:

$$t + r = r + t, \tag{1}$$

and we shall use Klaus Grue’s Logiweb system to check our proof. Cf. [1]. However, we use $[x]$ and $[y]$ as our variables.

During the course we have been introduced to mathematical logic using the text book *Introduction to Mathematical Logic* by Elliot Mendelson, cf. [4]. In order to get the full of this report we thus assume an elementary knowledge of logic as it is introduced in this book. However, we do give short introductions to some basic topics—Logiweb as well as logic—making this report selfcontained—at least to a graduate student of computer science at DIKU.

The report has been written from June 23rd to July 3rd. Due to this short period of time the proof is carried out as simpel as possible discarding any fancy solutions even before they have been fully conceived of. The commutativity of addition is part of Proposition 3.2 in [4], p. 156, which is proved immediately afterwards on pp. 157f. We follow this proof backchaining from the actual proof of (1) through other results of the proposition and necessary lemmas and theorems contained in the book. Besides Mendelson’s propositions we only make use of two tautologies and one lemma in the final proof.

We hope our intentions have been reached and that this report carries just a little of the fun it has been working with Logiweb. As Klaus always invites the Logiweb user to we do also: “Have fun”.

2 The Logiweb System

In this section we give a short introduction to the Logiweb system, a web-based system for distribution of mathematical definitions, lemmas, and proofs writing web-pages. Cf. [1]. For further details and elaboration on the myriad of intricacies and topics concerning the system we refer to Klaus Grue’s introduction to Logiweb, namely the [check](#) page. Alternatively, we recommend the [peano](#) page introducing peano arithmetic, which is displays well the system “at work”. In order to obtain a thorough knowledge of the system one is adviced to follow the course “Logic” or read the [base](#) page (cf. [2]), but probably the best choice would be to devour the exercise of writing a page proving some proposition of choice chosen from [4] using Logiweb. After such an exercise any logiweb work and the hardship of doing so will most likely depend on the level of mathematics one is working with.

2.1 References

As hinted at above Logiweb is a system in which users world wide can distribute their own mathematical work and reuse the work of others. Mathematics in Logiweb is publicized by submitting a *logiweb page* to the Logiweb system, that is, to a *Logiweb server*. Abstractly speaking, in a logiweb page one can define a mathematical theory or system¹ and a language of this system within which one can then define, state, and prove mathematical lemmas, theorems, propositions, etc. A theory must defined in one page. The proofs within a theory will be proof checked upon submission (actually they will be proof checked when compiling the page locally) to the Logiweb system and one then knows if the contents of a page are correct within the theory. A page can refer to other logiweb pages in which a needed theory, results, functionalities etc. are defined. Thus, Logiweb works much like the world wide web where pages refer to each other using hyperlinks. The mesh of Logiweb servers translate Logiweb references to http urls making sure a page can find its references and that they together form a directed acyclic graph from top (the base) to bottom (a logiweb page).

Essentially, the Logiweb reference of a page is global hash key computed on the basis of the contents of the specific page. This ensures that each page has a unique key, that is, reference. The reference of this page in the special *kana*² format is:

```
nani
sika nuta tita nate kesa   suke nesu nute tese kaku
nuki teni sike kute nenu   keti kanu sane suke suta
sasusaki seni keni kuke   sati sika nasa natu
```

Kana is a format developed for Logiweb making it possible to speak or pronounce a reference. This is done because Logiweb has been developed with the possibility of using a microphone as input tool in mind.

Also, one can refer to this page using the well-known http url:

```
http://www.diku.dk/ grue/logiweb/20050502/home/
mrmr/peano-commutativity/latest/vector/page.lgw
```

Two last ways to refer to a page is using the reference expressed in so-called *mixed endian hexadecimal* or using a decimal reference. Cf. [1] and the **Reference** section in the source or **pyk** code in a Logiweb page. The reference section—known as the **BIBLIOGRAPHY**—of a Logiweb page will be introduced shortly.

2.2 The Base Page

Now, in the previous section we mentioned *the base* of the mesh of Logiweb pages. This is a certain part of Logiweb, which implements the basic and necessary functionalities of the mathematical part of the Logiweb system. That is, the

¹We shall use *theory* and *system* interchangeably, as is the case with *Logiweb system* and *Logiweb*.

²For details on kana we refer to <http://yoa.dk/logiweb/doc/misc/kana.html>.

base page—as it is denoted—defines the proof checker and most of what makes the mathematics of Logiweb possible. Actually, there is possibly more than one base page, but the one Klaus Grue has written will probably be *the* base page to begin with. But one is free to write a base page making up a different (or a similar) system. A base page is a page without references to other pages and thus form the base of a mathematical system within Logiweb. For more on this we refer to the **base** page, Section 1.4 in particular.

We now know the necessary basics of the Logiweb system. Next, we take a closer look at the structure of a Logiweb page.

3 A Logiweb Page

In this section we shortly describe the most important details of a Logiweb page. That is, we introduce the Logiweb page, the language used in Logiweb, and the predefined structure all pages must be written according to. This introduction is adequate for getting started with Logiweb, but is in no manner satisfactory to the serious user of the Logiweb system.

3.1 “Main Menu”

In the Logiweb system each page has a so-called *main menu*, e.g., the main menu of this page is at:

<http://www.diku.dk/~grue/logiweb/20050502/home/mrmr/peano-commutativity/latest/>

From the main menu one can browse and find numerous kinds of information about the page. The most important of which to begin with is the page in **pdf** format and (for Logiweb newbie) the source code.

A Logiweb page is written in mixture of \LaTeX and the language **pyk** developed by Klaus Grue. **pyk** is then the language in which the mathematics to be checked by a computer of a Logiweb page is written. The best way to get a feeling for this mixture is to read the source code of a Logiweb page. This is found under **Source** \rightarrow **Actual Source** from the Logiweb main menu of a Logiweb page, e.g., the **check** page.

One can also read the **pyk** text of a page, which is found under **Body** \rightarrow **Pyk** also at the Logiweb main menu. It is also under **Body** that one finds the \TeX and pdf files of the page.³

3.2 The Pyk Language

As mentioned in the previous section the **pyk** language has been developed for easy pronunciation or in general for “spoken mathematics”, cf. the **base** page,

³The \TeX files are not available through a browser due to safety precautions.

Section 1.1. Thus, a mathematical expression as we are well-acquainted with such as:

$$(x + y)^2 = x^2 + 2xy + y^2$$

may be the equivalent of the following `pyk` expression:

```
parenthesis var x plus var y end parenthesis square equals var x
square plus two times var x times var y plus var y square4
```

Thus, in time one can “tell” one’s mathematics to a computer and it will generate a file written in `pyk`, which will then be the source for precisely one Logiweb page. This is also the reason why `pyk` is case sensitive, that is, the `pyk` code cannot be in capital letters.

We thus see that `pyk` is much like any other programming language. However, in `pyk` one writes everything out in words instead of using symbols, e.g., “parenthesis” and “end parenthesis” instead of “(” and “)”. In pdf format the textual representations will be replaced by their symbolic representations defined by the author of a Logiweb page. We shall return to this *aspect* of Logiweb in Section 4.2.

3.3 The Structure of a Logiweb Page

A Logiweb page has to be written according to certain standards. Thus, a specific structural sketch for a Logiweb page has to be used, which defines the specific and necessary parts of a Logiweb source file, that is, a `pyk` file. A `pyk` file has exactly one of each of these parts or sections, which will be described below. We note that the section names must be in capital letters. Like the \TeX code and normal text written this is not part of `pyk` and does not need to be in small letters.

PAGE A Logiweb page has a name which is given in the **PAGE** section of the `pyk` file. For instance, in the source code for this page one will find the **PAGE** section at the very top where it is seen that this page is named “peano commutativity”.

BIBLIOGRAPHY The references of a Logiweb page are given in the **BIBLIOGRAPHY** section of the `pyk` file. A page can refer to every possible Logiweb page, also if one of these pages refer to once referenced pages. The Logiweb server will make sure no cyclic references occur.

ASSOCIATIVITY In the **ASSOCIATIVITY** section of the `pyk` file the constructs of a logiweb page and their associativity—either **POST-** or **PRE-** are the options— are declared. The order of the constructs furthermore define their priority. Constructs declared with comma separation after the same associativity statement have same priority. (In a sense the *constructs* of a Logiweb page make up the available `pyk` code in a page and pages referring to the page declaring and defining the constructs.) Thus, we get that:

⁴This example has been taken from [2].

```
PREASSOCIATIVE * plus *, * minus *
```

gives that, e.g., $[x + y]$ have the same priority as $[x - y]$, where as:

```
PREASSOCIATIVE * times *
:
PREASSOCIATIVE * plus *
```

gives that, e.g., $[x \cdot y]$ have higher priority than $[x + y]$. These examples are from the **base** page. In order to use these constructs in another page they should be entered in the associativity section of the page in question with **base** written after the associativity assignment. The usage of the names of constructs from a page in the associativity section of another page is to place the priority of new constructs in a proper hierarcical order. Thus, in the associativity section this page we put names of, e.g., the lemmas to be proved together with the **bracket** declaration, which gives those constructs the same priority:

```
PREASSOCIATIVE base: bracket * end bracket,
mendelson lemma one eight eight
```

BODY In the **BODY** section we enter the actual text and math of the page. Hence, this is the section most readers get acquainted with, and they probably do so in pdf format. It is also in this section that the actual **pyk** coding is done.

We thus get the following preamble for a Logiweb page:

```
PAGE name
:
:
BIBLIOGRAPHY
base: nani
...
:
:
PREASSOCIATIVITY
:
:
BODY
:
:
```

The above only gives a bland taste of the richness of a very complex system that Logiweb is. But as stated earlier we believe it should be enough to get the reader started.

Now we proceed to the actual mathematics of this page, namely peano arithmetic. But before getting into details with this we place the assignment in a larger frame of reference.

4 Axiomatic Theories

In this section we give a basic introduction to the concepts with which we will be working in the rest of the report. It may well be skipped by the (experienced) logician.

For simple systems of logic we can decide the truth value of a statement by mere use of truth tables. Rather quickly this is not enough when expressing more complex systems of logic. Though it took our ancestors a couple of millennia to give us the formalistic tools to express only the simplest of logical statements the basic elements of the so-called formal axiomatic theories are quite simple. Cf. [4], pp. 34ff.

The branch of logic known as *propositional calculus* is simple to understand and serves well when introducing the concept of axiomatic theories. The abstract concept of propositions, i.e. the abstract denotation of sentences like “Jones is communist” and “Jones is an atheist”, and a few logical connectives, e.g. \neg and \Rightarrow , can be put together with one of more rules of inference in a proper way and thus form a calculus. The combined logical statements can express many types of statements, the truth of which we can determine by use of truth tables. We extend these concepts in the following manner. We define a formal theory \mathcal{L} when the following conditions are satisfied:⁵

1. In \mathcal{L} a countable set of symbols are given, a finite sequence of which is called an *expression* of \mathcal{L} .
2. A subset of the set of expressions of \mathcal{L} are called the set of *well-formed formulas* of \mathcal{L} or *wfs* for short.
3. A set of the wfs are called the *axioms* of \mathcal{L} . If it can effectively be decided whether a wf is an axiom then \mathcal{L} is said to be an *axiomatic* theory.⁶
4. A finite set of relations, R_1, \dots, R_n , among the wfs make up the *rules of inference*. We say that the wf \mathcal{B} is a *direct consequence* of a given set of wfs and that it is so by virtue of R_i , if for each R_i , there is a unique positive integer j such that, for every set of j wfs and each \mathcal{B} , one can effectively decide whether the given j wfs are in the relation R_i to \mathcal{B} .

A *proof* in \mathcal{L} is a sequence $\mathcal{B}_1, \dots, \mathcal{B}_k$ of wfs such that, for each i , either \mathcal{B}_i is an axiom of \mathcal{L} or \mathcal{B}_i is a direct consequence of some of the preceding wfs in the sequence by virtue of one of the rules of inference of \mathcal{L} .

A *theorem* of \mathcal{L} is a wf \mathcal{B} such that \mathcal{B} is the last wf of some proof in \mathcal{L} . Such a proof is called a *proof of \mathcal{B} in \mathcal{L}* . If it can be machine checked whether a given theorem is indeed a theorem of a given theory then we say that the theory is *decidable*, otherwise it is said to be *undecidable*.

We say that \mathcal{C} is a *consequence* of a set of wfs, Γ , if and only if there is a sequence of wfs, $\mathcal{B}_1, \dots, \mathcal{B}_k$, such that \mathcal{C} is \mathcal{B}_k , and, for each i , either \mathcal{B}_i is an

⁵We only state the most necessary concepts here, cf. [4], pp. 34f. for an elaborate explanation.

⁶We thus have that theories defined in Logiweb must be axiomatic theories.

axiom or \mathcal{B}_i is a direct consequence by some rule of inference of some of the preceding wfs in the sequence, which is then called a *proof* or (*deduction*) of \mathcal{C} from Γ . The members of Γ are called the *hypotheses* or *premises* of the proof.

We now have the prerequisites in order to define an actual theory.

4.1 The Axiomatic Theory L

The concept of a formal axiomatic theory L for the propositional calculus can now be introduced:

1. The symbols of L are $\neg, \Rightarrow, (,),$ and the letters $A_i,$ for $i \in \mathbb{N}$.
2. Well-formed formulas:
 - (a) All statement letters are wfs.
 - (b) if \mathcal{B} and \mathcal{C} are wfs, then so are $(\neg\mathcal{B})$ and $(\mathcal{B} \Rightarrow \mathcal{C})$.
3. If $\mathcal{B}, \mathcal{C},$ and \mathcal{D} are wfs of L, then the following are axioms of L:
 - (A1) $(\mathcal{B} \Rightarrow (\mathcal{C} \Rightarrow \mathcal{B}))$
 - (A2) $((\mathcal{B} \Rightarrow (\mathcal{C} \Rightarrow \mathcal{D})) \Rightarrow ((\mathcal{B} \Rightarrow \mathcal{C}) \Rightarrow (\mathcal{B} \Rightarrow \mathcal{D})))$
 - (A3) $((\neg\mathcal{C}) \Rightarrow (\neg\mathcal{B})) \Rightarrow (((\neg\mathcal{C}) \Rightarrow \mathcal{B}) \Rightarrow \mathcal{C}))$
4. One rule of inference of L, namely *modus ponens*, which says that \mathcal{C} is a direct consequence of \mathcal{B} and $\mathcal{B} \Rightarrow \mathcal{C}$. This rule is abbreviated by MP.

This theory can, of course, be elaborated upon, but we shall not pursue the details of this any further. Nor shall we make more distinctions in the logical concepts with which this assignment is concerned. Instead we shall place it in the Logiweb frame of reference.

4.2 An Axiomatic Logiweb Theory

It is in the conceptual framework introduced in the previous subsection that we shall work in when proving the law commutativity of addition within peano arithmetic. Since this proof must be carried out within the system of Logiweb we here give a Logiweb example and prove a very basic lemma from [4], p. 36, namely [M Lemma 1.8]⁷. (We shall adopt the convention of writing “M” in front of a lemma, proposition etc. to denote “Mendelson”).

Now, the appearance of “M Lemma 1.8” above is somewhat special. First, the name of the lemma (actually, it is not a lemma until defined to be so—now it is merely a *construct* in this page) appeared in brackets and with a footnote. This is specific to Logiweb and from it we are introduced to some of the *aspects* of Logiweb. Cf. the [base](#) page, Section 2.2.1.

⁷[M Lemma 1.8 $\stackrel{\text{pyk}}{=}$ “mendelson lemma one eight”]

If reading the **pyk** file one can find the declaration of [M Lemma 1.8] as being part of this Logiweb page almost at the very top, namely in the associativity section. And at this point in the text we introduce the construct to Logiweb, that is, from now on one can refer to the construct by writing some specific **pyk** code. One refers to this construct by writing what appears in the footnote of the construct after the equality sign with “pyk” written above it; this is an example of the *pyk aspect* of Logiweb. As is clear now, it simply defines the **pyk** code.

Another aspect is the *T_EX aspect* of Logiweb, which is simply how a construct in Logiweb appears in writing, that is, in those formats one can build from a normal T_EX document, e.g., a pdf document. The T_EX aspect is defined by the user or author who within the limits of T_EX single-handedly decides how the construct should appear.

Having made the construct visible or available to Logiweb, we want to place it some specific framework within which we work, that is, in a theory or system. We could denote this system L in accordance with what has been said above and build our theory from the bottom. Instead we shall make use of the work of Klaus Grue. Thus, we introduce [M Lemma 1.8] in [S’], which is defined on the **peano** page. We do this as follows:

[S’ **lemma** M Lemma 1.8: $\forall \mathcal{B}: \mathcal{B} \Rightarrow \mathcal{B}$]

For now we merely mention in passing that the special symbol \Rightarrow with the little dot appearing above the imply symbol tells us that this implication is specific to the peano arithmetic defined in [S’]. This system will be the one within which we will be working, and we shall examine it more in the next section.

We have now defined the construct [M Lemma 1.8] to be a lemma in the system [S’]. Then we must prove it in order to use it later should we have the need for that. This is done using axiomatic constructs from the peano page. For now, we shall use these without further introduction, which we defer until the next section:

S’ **proof of** M Lemma 1.8:

L01:	Arbitrary \gg	\mathcal{B}	;
L02:	A2’ \gg	$(\mathcal{B} \Rightarrow ((\mathcal{B} \Rightarrow \mathcal{B})) \Rightarrow \mathcal{B}) \Rightarrow$	
		$((\mathcal{B} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{B})) \Rightarrow (\mathcal{B} \Rightarrow \mathcal{B}))$;
L03:	A1’ \gg	$\mathcal{B} \Rightarrow ((\mathcal{B} \Rightarrow \mathcal{B}) \Rightarrow \mathcal{B})$;
L04:	MP’ \triangleright L02 \triangleright L03 \gg	$(\mathcal{B} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{B})) \Rightarrow (\mathcal{B} \Rightarrow \mathcal{B})$;
L05:	A1’ \gg	$\mathcal{B} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{B})$;
L06:	MP’ \triangleright L04 \triangleright L05 \gg	$\mathcal{B} \Rightarrow \mathcal{B}$	□

We have now seen an example of a proof of a lemma belonging to a certain theory in Logiweb. In the next section we will take a closer look at the axiomatic theory [S’] in which we will be working.

5 Peano Arithmetic

As stated above the assignment of this report is to prove the law of commutativity of addition within the natural numbers. More precisely, we shall prove it within the system of *peano arithmetic* as it is stated in Chapter 3 of [4]. Peano arithmetic is based upon five postulates:

- (P1) 0 is a natural number.
- (P2) If x is a natural number, then there is another natural number denoted by x' ; this is denoted the *successor* of x .)
- (P3) $0 \neq x'$ for every natural number x .
- (P4) If $x' = y'$, then $x = y$.
- (P5) If Q is a property that may or may not hold for any given natural number, and if:
 - (I) 0 has the property Q , and
 - (II) whenever a natural number x has the property Q , then x' has the property Q ,

then all natural numbers have the property Q . This is the mathematical principle of induction.

Klaus Grue has implemented peano arithmetic making most of the needed constructs to solve the assignment available to us. Cf. [peano](#), Section 1.5. The result of his work is among other things the theory within which we will be working, namely the aforementioned theory [S']. This theory is much like the axiomatic theory we met in Section 4.1, p. 9. Only, this theory has more axioms, a number of theorems we can use, and one more rule of inference. We repeat the full system in the following for easy reference.

5.1 Peano Arithmetic in Logiweb

Peano arithmetic in Logiweb consists of the axioms, theorems, and rules of inference that is restated in the following subsections; they are introduced and defined in [3]. The appearance of the constructs is somewhat special. As mentioned above some symbols are furnished with a little dot. This special \TeX aspect need not confuse us. It only shows that we are working within peano arithmetic as defined in the [peano](#) page. Also, the equality symbol of peano arithmetic is furnished with a small p , which serves the same purpose as the dot.

5.1.1 S' Axioms

In peano arithmetic—or in [Theory S'] or merely [S'] as we shall refer to it from now on—the following five axioms or rules are available.

$$[\text{S' rule A1'}: \forall \mathcal{A}: \forall \mathcal{B}: \mathcal{A} \Rightarrow \mathcal{B} \Rightarrow \mathcal{A}]$$

$$[\text{S' rule A2'}: \forall \mathcal{A}: \forall \mathcal{B}: \forall \mathcal{C}: (\mathcal{A} \Rightarrow \mathcal{B} \Rightarrow \mathcal{C}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \mathcal{A} \Rightarrow \mathcal{C}]$$

$$[\text{S' rule A3'}: \forall \mathcal{A}: \forall \mathcal{B}: (\dot{\neg} \mathcal{B} \Rightarrow \dot{\neg} \mathcal{A}) \Rightarrow (\dot{\neg} \mathcal{B} \Rightarrow \mathcal{A}) \Rightarrow \mathcal{B}]$$

$$[\text{S' rule A4'}: \forall \mathcal{C}: \forall \mathcal{A}: \forall \mathcal{X}: \forall \mathcal{B}: [\mathcal{A}] \equiv \langle [\mathcal{B}] \mid [\mathcal{X}] := [\mathcal{C}] \rangle \Vdash \dot{\forall} \mathcal{X}: \mathcal{B} \Rightarrow \mathcal{A}]$$

$$[\text{S' rule A5'}: \forall \mathcal{X}: \forall \mathcal{A}: \forall \mathcal{B}: \text{nonfree}([\mathcal{X}], [\mathcal{A}]) \Vdash \dot{\forall} \mathcal{X}: (\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \mathcal{A} \Rightarrow \dot{\forall} \mathcal{X}: \mathcal{B}]$$

These definitions are copied from the [peano](#) page, Section 1.5.

5.1.2 S' Rules of Inference

Besides the five axioms [Theory S'] has two rules of inference. The first is the well-known modus ponens, which all computer scientists are well-acquainted with. The second says that a well-formed formula has as a consequence that it is sound or valid for all variables, x_i .

$$[\text{S' rule MP'}: \forall \mathcal{A}: \forall \mathcal{B}: \mathcal{A} \Rightarrow \mathcal{B} \vdash \mathcal{A} \vdash \mathcal{B}]$$

$$[\text{S' rule Gen'}: \forall \mathcal{X}: \forall \mathcal{A}: \mathcal{A} \vdash \dot{\forall} \mathcal{X}: \mathcal{A}]$$

These definitions are copied from the [peano](#) page, Section 1.5.

5.1.3 S' Theorems

Finally, within [Theory S'] we are allowed to use the following theorems. We shall not prove these:

$$[\text{S' rule S1'}: \forall \mathcal{A}: \forall \mathcal{B}: \forall \mathcal{C}: \mathcal{A} \stackrel{\text{p}}{=} \mathcal{B} \Rightarrow \mathcal{A} \stackrel{\text{p}}{=} \mathcal{C} \Rightarrow \mathcal{B} \stackrel{\text{p}}{=} \mathcal{C}]$$

$$[\text{S' rule S2'}: \forall \mathcal{A}: \forall \mathcal{B}: \mathcal{A} \stackrel{\text{p}}{=} \mathcal{B} \Rightarrow \mathcal{A}' \stackrel{\text{p}}{=} \mathcal{B}']$$

$$[\text{S' rule S3'}: \forall \mathcal{A}: \neg \dot{0} \stackrel{\text{p}}{=} \mathcal{A}']$$

$$[\text{S' rule S4'}: \forall \mathcal{A}: \forall \mathcal{B}: \mathcal{A}' \stackrel{\text{p}}{=} \mathcal{B}' \Rightarrow \mathcal{A} \stackrel{\text{p}}{=} \mathcal{B}]$$

$$[\text{S' rule S5'}: \forall \mathcal{A}: \mathcal{A} \dot{+} \dot{0} \stackrel{\text{p}}{=} \mathcal{A}]$$

$$[\text{S' rule S6'}: \forall \mathcal{A}: \forall \mathcal{B}: \mathcal{A} \dot{+} \mathcal{B}' \stackrel{\text{p}}{=} (\mathcal{A} \dot{+} \mathcal{B})']$$

$$[\text{S' rule S7'}: \forall \mathcal{A}: \mathcal{A} \dot{:} \dot{0} \stackrel{\text{p}}{=} \dot{0}]$$

$$[S' \text{ rule } S8': \forall \mathcal{A}: \forall \mathcal{B}: \mathcal{A} : (\mathcal{B}') \stackrel{P}{=} (\mathcal{A} : \mathcal{B}) \dot{+} \mathcal{A}]$$

$$\begin{aligned} [S' \text{ rule } S9': \forall \mathcal{A}: \forall \mathcal{B}: \forall \mathcal{C}: \forall \mathcal{X}: \\ \mathcal{B} \equiv \langle \mathcal{A} | \mathcal{X} := \dot{0} \rangle \vdash \mathcal{C} \equiv \langle \mathcal{A} | \mathcal{X} := \mathcal{X}' \rangle \vdash \\ \mathcal{B} \Rightarrow \forall \mathcal{X}: (\mathcal{A} \Rightarrow \mathcal{C}) \Rightarrow \forall \mathcal{X}: \mathcal{A}] \end{aligned}$$

These definitions are copied from the [peano](#) page, Section 1.5.

We now have the necessary means to solve our assignment, which we will do in the following sections.

6 Proof of Commutativity of Addition

As mentioned in the introduction we shall prove the law of commutativity of addition in as simple a way as possible. We thus follow Mendelson in his proof, cf. [4], pp. 156ff. Hence, we will prove this law by proving Proposition 3.2(a)-(h), except Proposition 3.2(e). As we close in on our goal we will be needing two tautologies and in our last proof a lemma, which we will state and prove as we go along. We do this in the following subsections. We explain what goes on as we go along. In each proof we first introduce the proposition in Logiweb, then we state the proposition, and finally we prove it.

6.1 M Proposition 3.2(a)

The first proposition states that a wf is equal to itself. Since peano arithmetic is theory with equality this is basic and expected proposition—and quite logical!

[M Proposition 3.2(a)]⁸

[S' lemma M Proposition 3.2(a): $\forall \mathcal{T}: \mathcal{T} \stackrel{P}{=} \mathcal{T}$]

S' **proof of** M Proposition 3.2(a):

L01:	Arbitrary \gg	\mathcal{T}	;
L02:	S5' \gg	$\mathcal{T} \dot{+} \dot{0} \stackrel{P}{=} \mathcal{T}$;
L03:	S1' \gg	$(\mathcal{T} \dot{+} \dot{0} \stackrel{P}{=} \mathcal{T}) \Rightarrow (\mathcal{T} \dot{+} \dot{0} \stackrel{P}{=} \mathcal{T} \Rightarrow$	
		$\mathcal{T} \stackrel{P}{=} \mathcal{T})$;
L04:	MP' \triangleright L03 \triangleright L02 \gg	$\mathcal{T} \dot{+} \dot{0} \stackrel{P}{=} \mathcal{T} \Rightarrow \mathcal{T} \stackrel{P}{=} \mathcal{T}$;
L05:	MP' \triangleright L04 \triangleright L02 \gg	$\mathcal{T} \stackrel{P}{=} \mathcal{T}$	□

Proposition 3.2(a) is used in the proof of Proposition 3.2(b). But before we can proceed to this we must prove a tautology also.

⁸[M Proposition 3.2(a)]^{pyk} $\stackrel{pyk}{=} \text{“mendelson proposition three two a”}$

6.2 M Tautology A

In [4], p. 157 in the proof of Proposition 3.2(b) we see that a tautology is used. We shall state and prove this tautology for use in our next and later proofs.

[M Tautology A]⁹

[S' lemma M Tautology A: $\forall \mathcal{A}: \forall \mathcal{B}: \forall \mathcal{C}: \mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C}) \vdash \mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$]

S' proof of M Tautology A:

L01:	Arbitrary \gg	\mathcal{A}	;
L02:	Arbitrary \gg	\mathcal{B}	;
L03:	Arbitrary \gg	\mathcal{C}	;
L04:	Premise \gg	$\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})$;
L05:	A2' \gg	$(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$;
L06:	MP' \triangleright L05 \triangleright L04 \gg	$(\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$;
L07:	A1' \gg	$((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B})) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$;
L08:	MP' \triangleright L07 \triangleright L06 \gg	$\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$;
L09:	A2' \gg	$(\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B})) \Rightarrow (\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$;
L10:	MP' \triangleright L09 \triangleright L08 \gg	$(\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B})) \Rightarrow (\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$;
L11:	A1' \gg	$\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B})$;
L12:	MP' \triangleright L10 \triangleright L11 \gg	$\mathcal{B} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$	□

We note that [M Proposition 3.2(b)] could also be proved by the help of Corollary 1.10(b), cf. [4], p. 38. Mendelson proves this by the use of Proposition 1.9 also know as the Deduction Theorem, cf. [4], p. 37. We avoid the use of this, though, it would have been prettier to follow Mendelson completely.

6.3 M Proposition 3.2(b)

We now proceed to proving [M Proposition 3.2(b)] where we will make use of [M Tautology A].

[M Proposition 3.2(b)]¹⁰

[S' lemma M Proposition 3.2(b): $\forall \mathcal{T}: \forall \mathcal{R}: \mathcal{T} \stackrel{\text{p}}{=} \mathcal{R} \Rightarrow \mathcal{R} \stackrel{\text{p}}{=} \mathcal{T}$]

S' proof of M Proposition 3.2(b):

L01:	Arbitrary \gg	\mathcal{T}	;
------	-----------------	---------------	---

⁹[M Tautology A $\stackrel{\text{pyk}}{=} \text{"mendelson tautology a"}]$

¹⁰[M Proposition 3.2(b) $\stackrel{\text{pyk}}{=} \text{"mendelson proposition three two b"}]$

L02:	Arbitrary \gg	\mathcal{R}	;
L03:	S1' \gg	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \Rightarrow (\mathcal{T} \stackrel{p}{=} \mathcal{T} \Rightarrow \mathcal{R} \stackrel{p}{=} \mathcal{T})$;
L04:	M Tautology A \triangleright L03 \gg	$\mathcal{T} \stackrel{p}{=} \mathcal{T} \Rightarrow (\mathcal{T} \stackrel{p}{=} \mathcal{R} \Rightarrow \mathcal{R} \stackrel{p}{=} \mathcal{T})$;
L05:	M Proposition 3.2(a) \gg	$\mathcal{T} \stackrel{p}{=} \mathcal{T}$;
L06:	MP' \triangleright L04 \triangleright L05 \gg	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \Rightarrow \mathcal{R} \stackrel{p}{=} \mathcal{T}$	□

Thus having proved [M Proposition 3.2(b)] we proceed to [M Proposition 3.2(c)] in the proof of which the former is used. But before we do this we need yet another tautology.

6.4 M Tautology B

In [4], p. 157 in the proof of Proposition 3.2(c) we see that a tautology is used. We shall state and prove this tautology for use in our next proof.

[M Tautology B]¹¹

[S' lemma M Tautology B: $\forall \mathcal{D}: \forall \mathcal{E}: \forall \mathcal{F}: \mathcal{D} \Rightarrow \mathcal{E} \vdash \mathcal{E} \Rightarrow \mathcal{F} \vdash \mathcal{D} \Rightarrow \mathcal{F}$]

S' proof of M Tautology B:

L01:	Arbitrary \gg	\mathcal{D}	;
L02:	Arbitrary \gg	\mathcal{E}	;
L03:	Arbitrary \gg	\mathcal{F}	;
L04:	Premise \gg	$\mathcal{D} \Rightarrow \mathcal{E}$;
L05:	Premise \gg	$\mathcal{E} \Rightarrow \mathcal{F}$;
L06:	A1' \gg	$(\mathcal{E} \Rightarrow \mathcal{F}) \Rightarrow (\mathcal{D} \Rightarrow (\mathcal{E} \Rightarrow \mathcal{F}))$;
L07:	MP' \triangleright L06 \triangleright L05 \gg	$\mathcal{D} \Rightarrow (\mathcal{E} \Rightarrow \mathcal{F})$;
L08:	A2' \gg	$(\mathcal{D} \Rightarrow (\mathcal{E} \Rightarrow \mathcal{F})) \Rightarrow ((\mathcal{D} \Rightarrow \mathcal{E}) \Rightarrow (\mathcal{D} \Rightarrow \mathcal{F}))$;
L09:	MP' \triangleright L08 \triangleright L07 \gg	$(\mathcal{D} \Rightarrow \mathcal{E}) \Rightarrow (\mathcal{D} \Rightarrow \mathcal{F})$;
L10:	MP' \triangleright L09 \triangleright L04 \gg	$\mathcal{D} \Rightarrow \mathcal{F}$	□

We can now prove [M Proposition 3.2(c)].

6.5 M Proposition 3.2(c)

As noted above we will make use of [M Tautology B] in our proof of [M Proposition 3.2(c)] which concerns transitivity.

[M Proposition 3.2(c)]¹²

[S' lemma M Proposition 3.2(c): $\forall \mathcal{T}: \forall \mathcal{R}: \forall \mathcal{S}: \mathcal{T} \stackrel{p}{=} \mathcal{R} \Rightarrow (\mathcal{R} \stackrel{p}{=} \mathcal{S} \Rightarrow \mathcal{T} \stackrel{p}{=} \mathcal{S})$]

¹¹[M Tautology B $\stackrel{\text{pyk}}{=} \text{"mendelson tautology b"}]$

¹²[M Proposition 3.2(c) $\stackrel{\text{pyk}}{=} \text{"mendelson proposition three two c"}]$

S' proof of M Proposition 3.2(c):

L01:	Arbitrary \gg	\mathcal{T}	;
L02:	Arbitrary \gg	\mathcal{R}	;
L03:	Arbitrary \gg	\mathcal{S}	;
L04:	$\mathcal{S}1' \gg$	$\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow (\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S} \Rightarrow \mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{S})$;
L05:	M Proposition 3.2(b) \gg	$\mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{R} \Rightarrow \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T}$;
L06:	M Tautology B \triangleright L05 \triangleright L04 \gg	$\mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{R} \Rightarrow (\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S} \Rightarrow \mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{S})$	□

Thus having proved [M Proposition 3.2(c)] we now proceed with [M Proposition 3.2(d)]

6.6 M Proposition 3.2(d)

In order to prove [M Proposition 3.2(d)] following the proof in [4], p. 157, we will be using two tautologies. A closer examination of the proofs reveals that the two tautologies are an instance of [M Tautology A] and one of [M Tautology A]. Furthermore, we will be needing [M Proposition 3.2(b)] in our proof as well. Hence, we can commence with the proof right away.

[M Proposition 3.2(d)]¹³

[S' lemma M Proposition 3.2(d): $\forall \mathcal{R}: \forall \mathcal{T}: \forall \mathcal{S}: \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow (\mathcal{S} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S})$]

S' proof of M Proposition 3.2(d):

L01:	Arbitrary \gg	\mathcal{R}	;
L02:	Arbitrary \gg	\mathcal{T}	;
L03:	Arbitrary \gg	\mathcal{S}	;
L04:	M Proposition 3.2(c) \gg	$\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow (\mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{S} \Rightarrow \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S})$;
L05:	M Tautology A \triangleright L04 \gg	$\mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{S} \Rightarrow (\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S})$;
L06:	M Proposition 3.2(b) \gg	$\mathcal{S} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow \mathcal{T} \stackrel{\mathcal{P}}{=} \mathcal{S}$;
L07:	M Tautology B \triangleright L06 \triangleright L05 \gg	$\mathcal{S} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow (\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S})$;
L08:	M Tautology A \triangleright L07 \gg	$\mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow (\mathcal{S} \stackrel{\mathcal{P}}{=} \mathcal{T} \Rightarrow \mathcal{R} \stackrel{\mathcal{P}}{=} \mathcal{S})$	□

Thus having proved [M Proposition 3.2(d)] we now proceed with [M Proposition 3.2(e)] in the proof of which we use the result just obtained.

6.7 M Proposition 3.2(f)

The proof of [M Proposition 3.2(f)] is split up into two parts as is seen from [4], pp. 157f.

6.7.1 M Proposition 3.2(f) (i)

After having introduced the proposition and placed it in the theory in which we are working, we first show the identity of $[\dot{0}]$. This is the base case of a

¹³[M Proposition 3.2(d)] $\stackrel{\text{pyk}}{=} \text{“mendelson proposition three two d”}$

well-known proof by induction. After have proved the base case, we shall thus proceed and prove the induction step by the use of our two tautologies. Finally, we prove [M Proposition 3.2(f)] by the use of our two intermediary results.

[M Proposition 3.2(f) (i)]¹⁴

[S' lemma M Proposition 3.2(f) (i): $\dot{0} \stackrel{P}{=} \dot{0} \dot{+} \dot{0}$]

S' proof of M Proposition 3.2(f) (i):

L01:	$S5' \gg$	$\dot{0} \dot{+} \dot{0} \stackrel{P}{=} \dot{0}$;
L02:	M Proposition 3.2(b) \gg	$\dot{0} \dot{+} \dot{0} \stackrel{P}{=} \dot{0} \Rightarrow \dot{0} \stackrel{P}{=} \dot{0} \dot{+} \dot{0}$;
L03:	$MP' \triangleright L02 \triangleright L01 \gg$	$\dot{0} \stackrel{P}{=} \dot{0} \dot{+} \dot{0}$	□

We have now showed the base case in an induction. The next step is to continue the induction.

6.7.2 M Proposition 3.2(f) (ii)

In Mendelson's proof of [M Proposition 3.2(f) (ii)] we note that the Deduction Theorem is used, cf. [4], p. 158. We shall not use this theorem in our proof but instead use the previously proved tautologies.

[M Proposition 3.2(f) (ii)]¹⁵

[S' lemma M Proposition 3.2(f) (ii): $\dot{\forall} \dot{x}: (\dot{x} \stackrel{P}{=} \dot{0} \dot{+} \dot{x} \Rightarrow \dot{x}' \stackrel{P}{=} \dot{0} \dot{+} \dot{x}')$]

S' proof of M Proposition 3.2(f) (ii):

L01:	$S6' \gg$	$\dot{0} \dot{+} \dot{x}' \stackrel{P}{=} (\dot{0} \dot{+} \dot{x})'$;
L02:	$S2' \gg$	$\dot{x} \stackrel{P}{=} \dot{0} \dot{+} \dot{x} \Rightarrow \dot{x}' \stackrel{P}{=} (\dot{0} \dot{+} \dot{x})'$;
L03:	M Proposition 3.2(d) \gg	$\dot{x}' \stackrel{P}{=} (\dot{0} \dot{+} \dot{x})' \Rightarrow (\dot{0} \dot{+} \dot{x}' \stackrel{P}{=} (\dot{0} \dot{+} \dot{x})' \Rightarrow \dot{x}' \stackrel{P}{=} \dot{0} \dot{+} \dot{x}')$;
L04:	M Tautology B $\triangleright L02 \triangleright L03 \gg$	$\dot{x} \stackrel{P}{=} \dot{0} \dot{+} \dot{x} \Rightarrow (\dot{0} \dot{+} \dot{x}' \stackrel{P}{=} (\dot{0} \dot{+} \dot{x})' \Rightarrow \dot{x}' \stackrel{P}{=} \dot{0} \dot{+} \dot{x}')$;
L05:	M Tautology A $\triangleright L04 \gg$	$\dot{0} \dot{+} \dot{x}' \stackrel{P}{=} (\dot{0} \dot{+} \dot{x})' \Rightarrow (\dot{x} \stackrel{P}{=} \dot{0} \dot{+} \dot{x} \Rightarrow \dot{x}' \stackrel{P}{=} \dot{0} \dot{+} \dot{x}')$;
L06:	$MP' \triangleright L05 \triangleright L01 \gg$	$\dot{x} \stackrel{P}{=} \dot{0} \dot{+} \dot{x} \Rightarrow \dot{x}' \stackrel{P}{=} \dot{0} \dot{+} \dot{x}'$;
L07:	$Gen' \triangleright L06 \gg$	$\dot{\forall} \dot{x}: (\dot{x} \stackrel{P}{=} \dot{0} \dot{+} \dot{x} \Rightarrow \dot{x}' \stackrel{P}{=} \dot{0} \dot{+} \dot{x}')$	□

We now have the necessary foundation to prove [M Proposition 3.2(f)].

¹⁴[M Proposition 3.2(f) (i)] ^{pyk} $\stackrel{pyk}{=}$ "mendelson proposition three two f i"]

¹⁵[M Proposition 3.2(f) (ii)] ^{pyk} $\stackrel{pyk}{=}$ "mendelson proposition three two f ii"]

6.7.3 M Proposition 3.2(f)

As is seen from [4], p. 158, induction is used in the proof of [M Proposition 3.2(f)]. Due to our ambition of keeping things as simple as possible, we have not implemented an induction theorem. Hence, the proof will deviate from the one given by Mendelson.

[M Proposition 3.2(f)]¹⁶

[S' lemma M Proposition 3.2(f): $\dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x)$]

S' **proof of** M Proposition 3.2(f):

L01:	S9' \gg	$(\dot{0} \stackrel{P}{=} \dot{0} \dot{+} \dot{0}) \Rightarrow \dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x \Rightarrow x' \stackrel{P}{=} \dot{0} \dot{+} x') \Rightarrow \dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x)$;
L02:	M Proposition 3.2(f) (i) \gg	$\dot{0} \stackrel{P}{=} \dot{0} \dot{+} \dot{0}$;
L03:	MP' \triangleright L01 \triangleright L02 \gg	$\dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x \Rightarrow x' \stackrel{P}{=} \dot{0} \dot{+} x') \Rightarrow \dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x)$;
L04:	M Proposition 3.2(f) (ii) \gg	$\dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x \Rightarrow x' \stackrel{P}{=} \dot{0} \dot{+} x')$;
L05:	MP' \triangleright L03 \triangleright L04 \gg	$\dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x)$ \square

We now proceed to proving [M Proposition 3.2(g)], which is used in our final proof.

6.8 M Proposition 3.2(g)

The proof of [M Proposition 3.2(g)] is very much like our previous proof. Also, in this case do we need a base case and an induction step.

6.8.1 M Proposition 3.2(g) (i)

After having introduced the proposition and placed it in the theory in which we are working, we first show the identity of $[\dot{0}]$.

[M Proposition 3.2(g) (i)]¹⁷

[S' lemma M Proposition 3.2(g) (i): $\dot{\forall}x: (x' \dot{+} \dot{0} \stackrel{P}{=} (x \dot{+} \dot{0})')$]

S' **proof of** M Proposition 3.2(g) (i):

L01:	S5' \gg	$x' \dot{+} \dot{0} \stackrel{P}{=} x'$;
L02:	S5' \gg	$x \dot{+} \dot{0} \stackrel{P}{=} x$;
L03:	S2' \gg	$x \dot{+} \dot{0} \stackrel{P}{=} x \Rightarrow (x \dot{+} \dot{0})' \stackrel{P}{=} x'$;
L04:	MP' \triangleright L03 \triangleright L02 \gg	$(x \dot{+} \dot{0})' \stackrel{P}{=} x'$;

¹⁶[M Proposition 3.2(f)] $\stackrel{\text{pyk}}{=} \text{"mendelson proposition three two f"}$

¹⁷[M Proposition 3.2(g) (i)] $\stackrel{\text{pyk}}{=} \text{"mendelson proposition three two g i"}$

L05:	M Proposition 3.2(d) \gg	$\dot{x}' + \dot{0} \stackrel{P}{=} \dot{x}' \Rightarrow (\dot{x} + \dot{0})' \stackrel{P}{=} \dot{x}' \Rightarrow$ $\dot{x}' + \dot{0} \stackrel{P}{=} (\dot{x} + \dot{0})'$;
L06:	MP' \triangleright L05 \triangleright L01 \gg	$(\dot{x} + \dot{0})' \stackrel{P}{=} \dot{x}' \Rightarrow \dot{x}' + \dot{0} \stackrel{P}{=}$ $(\dot{x} + \dot{0})'$;
L07:	MP' \triangleright L06 \triangleright L04 \gg	$\dot{x}' + \dot{0} \stackrel{P}{=} (\dot{x} + \dot{0})'$;
L08:	Gen' \triangleright L07 \gg	$\forall \dot{x}: (\dot{x}' + \dot{0} \stackrel{P}{=} (\dot{x} + \dot{0})')$	□

6.8.2 M Proposition 3.2(g) (ii)

We now proceed with the induction step.

[M Proposition 3.2(g) (ii)]¹⁸

[S' **lemma** M Proposition 3.2(g) (ii): $\forall \dot{y}: (\dot{x}' + \dot{y} \stackrel{P}{=} (\dot{x} + \dot{y})' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y}')')$]

S' **proof of** M Proposition 3.2(g) (ii):

L01:	S6' \gg	$\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x}' + \dot{y})'$;
L02:	S2' \gg	$\dot{x}' + \dot{y} \stackrel{P}{=} (\dot{x} + \dot{y})' \Rightarrow (\dot{x}' + \dot{y})' \stackrel{P}{=}$ $(\dot{x} + \dot{y})''$;
L03:	M Proposition 3.2(c) \gg	$\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x}' + \dot{y})' \Rightarrow$ $(\dot{x}' + \dot{y})' \stackrel{P}{=} (\dot{x} + \dot{y})'' \Rightarrow$ $\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y})''$;
L04:	M Tautology A \triangleright L03 \gg	$(\dot{x}' + \dot{y})' \stackrel{P}{=} (\dot{x} + \dot{y})'' \Rightarrow$ $\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x}' + \dot{y})' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=}$ $(\dot{x} + \dot{y})''$;
L05:	M Tautology B \triangleright L02 \triangleright L04 \gg	$\dot{x}' + \dot{y} \stackrel{P}{=} (\dot{x} + \dot{y})' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=}$ $(\dot{x}' + \dot{y})' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y})''$;
L06:	M Tautology A \triangleright L05 \gg	$\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x}' + \dot{y})' \Rightarrow \dot{x}' + \dot{y} \stackrel{P}{=}$ $(\dot{x} + \dot{y})' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y})''$;
L07:	MP' \triangleright L06 \triangleright L01 \gg	$\dot{x}' + \dot{y} \stackrel{P}{=} (\dot{x} + \dot{y})' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=}$ $(\dot{x} + \dot{y})''$;
L08:	S6' \gg	$\dot{x} + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y})'$;
L09:	S2' \gg	$\dot{x} + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y})' \Rightarrow (\dot{x} + \dot{y}')' \stackrel{P}{=}$ $(\dot{x} + \dot{y})''$;
L10:	M Proposition 3.2(d) \gg	$\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y})'' \Rightarrow$ $(\dot{x} + \dot{y}')' \stackrel{P}{=} (\dot{x} + \dot{y})'' \Rightarrow$ $\dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y}')'$;
L11:	M Tautology B \triangleright L07 \triangleright L10 \gg	$\dot{x}' + \dot{y} \stackrel{P}{=} (\dot{x} + \dot{y})' \Rightarrow (\dot{x} + \dot{y}')' \stackrel{P}{=}$ $(\dot{x} + \dot{y})'' \Rightarrow \dot{x}' + \dot{y}' \stackrel{P}{=} (\dot{x} + \dot{y}')'$;

¹⁸[M Proposition 3.2(g) (ii)]^{pyk} “mendelson proposition three two g ii”]

L12:	M Tautology A \triangleright L11 \gg	$(\dot{x} \dot{+} \dot{y}')' \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})'' \Rightarrow$ $\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})' \Rightarrow \dot{x}' \dot{+} \dot{y}' \stackrel{P}{=}$ $(\dot{x} \dot{+} \dot{y}')'$;
L13:	MP' \triangleright L09 \triangleright L08 \gg	$(\dot{x} \dot{+} \dot{y}')' \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})''$;
L14:	MP' \triangleright L12 \triangleright L13 \gg	$\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})' \Rightarrow \dot{x}' \dot{+} \dot{y}' \stackrel{P}{=}$ $(\dot{x} \dot{+} \dot{y}')'$;
L15:	Gen' \triangleright L14 \gg	$\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})') \Rightarrow$ $\dot{x}' \dot{+} \dot{y}' \stackrel{P}{=} (\dot{x} \dot{+} \dot{y}')'$ \square

6.8.3 M Proposition 3.2(g)

We now have the necessary means to prove [M Proposition 3.2(g)].

[M Proposition 3.2(g)]¹⁹

[S' lemma M Proposition 3.2(g): $\dot{\forall} \dot{y}: \dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})'$]

S' **proof of** M Proposition 3.2(g):

L01:	S9' \gg	$\dot{\forall} \dot{x}: (\dot{x}' \dot{+} \dot{0} \stackrel{P}{=} (\dot{x} \dot{+} \dot{0})') \Rightarrow$ $\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})') \Rightarrow$ $\dot{x}' \dot{+} \dot{y}' \stackrel{P}{=} (\dot{x} \dot{+} \dot{y}')' \Rightarrow$ $\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})')$;
L02:	M Proposition 3.2(g) (i) \gg	$\dot{\forall} \dot{x}: (\dot{x}' \dot{+} \dot{0} \stackrel{P}{=} (\dot{x} \dot{+} \dot{0})')$;
L03:	MP' \triangleright L01 \triangleright L02 \gg	$\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})') \Rightarrow$ $\dot{x}' \dot{+} \dot{y}' \stackrel{P}{=} (\dot{x} \dot{+} \dot{y}')' \Rightarrow$ $\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})')$;
L04:	M Proposition 3.2(g) (ii) \gg	$\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})') \Rightarrow$ $\dot{x}' \dot{+} \dot{y}' \stackrel{P}{=} (\dot{x} \dot{+} \dot{y}')'$;
L05:	MP' \triangleright L03 \triangleright L04 \gg	$\dot{\forall} \dot{y}: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})')$ \square

We have thus proved [M Proposition 3.2(g)].

6.9 M Proposition 3.2(h)

We now have the necessary means to show our objective, namely M Proposition 3.2(h). As was the case for M Proposition 3.2(f) and M Proposition 3.2(g) also this proposition will be shown by induction, that is, with a base case and an inductive step but without an induction theorem.

6.9.1 M Proposition 3.2(h)(i)

The proof of [M Proposition 3.2(f)] makes use of the @ operator, which is defined on the **base** page, Section 6.3.9. It works as a quantifier eliminator, that is,

¹⁹[M Proposition 3.2(g)] $\stackrel{\text{pyk}}{=} \text{“mendelson proposition three two g”}$

a sort of reverse of generalization. First, we prove the base case.

[M Proposition 3.2(h)(i)]²⁰

[S' lemma M Proposition 3.2(h)(i): $\dot{\forall}x: (x \dot{+} \dot{0} \stackrel{P}{=} \dot{0} \dot{+} x)$]

S' **proof of** M Proposition 3.2(h)(i):

L01:	S5' \gg	$\dot{x} \dot{+} \dot{0} \stackrel{P}{=} \dot{x}$;
L02:	M Proposition 3.2(f) \gg	$\dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x)$;
L03:	A4' @ $\dot{x} \gg$	$\dot{\forall}x: (x \stackrel{P}{=} \dot{0} \dot{+} x) \Rightarrow x \stackrel{P}{=} \dot{0} \dot{+} x$;
L04:	MP' \triangleright L03 \triangleright L02 \gg	$\dot{x} \stackrel{P}{=} \dot{0} \dot{+} x$;
L05:	M Proposition 3.2(c) \gg	$\dot{x} \dot{+} \dot{0} \stackrel{P}{=} \dot{x} \Rightarrow (x \stackrel{P}{=} \dot{0} \dot{+} x \Rightarrow$ $\dot{x} \dot{+} \dot{0} \stackrel{P}{=} \dot{0} \dot{+} x)$;
L06:	MP' \triangleright L05 \triangleright L01 \gg	$\dot{x} \stackrel{P}{=} \dot{0} \dot{+} x \Rightarrow \dot{x} \dot{+} \dot{0} \stackrel{P}{=} \dot{0} \dot{+} x$;
L07:	MP' \triangleright L06 \triangleright L04 \gg	$\dot{x} \dot{+} \dot{0} \stackrel{P}{=} \dot{0} \dot{+} x$;
L08:	Gen' \triangleright L07 \gg	$\dot{\forall}x: (x \dot{+} \dot{0} \stackrel{P}{=} \dot{0} \dot{+} x)$	□

Thus, we have proved the base case and now proceed to the induction step. But before we can prove this we need a lemma, which must be proved first.

6.9.2 M Proposition 3.2(h)_{(g)(ii)}

In the proof of [M Proposition 3.2(h)(ii)] line 3 we see that [M Proposition 3.2(g)] is used. Cf. [4], p. 158. In comparison to the proof given here the x and y variables are swaped. We thus need a lemma allowing this version of the proposition. This is because we have used constants in stating and proving the proposition.

The lemma we need is as follows:

[M Proposition 3.2(h)_{(g)(ii)}]²¹

[S' lemma M Proposition 3.2(h)_{(g)(ii)}: $\dot{\forall}y: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})') \vdash \dot{y}' \dot{+} \dot{x} \stackrel{P}{=} (\dot{y} \dot{+} \dot{x})'$]

S' **proof of** M Proposition 3.2(h)_{(g)(ii)}:

L01:	Premise \gg	$\dot{\forall}y: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})')$;
L02:	A4' @ $\dot{z} \gg$	$\dot{\forall}y: (\dot{x}' \dot{+} \dot{y} \stackrel{P}{=} (\dot{x} \dot{+} \dot{y})') \Rightarrow$ $\dot{x}' \dot{+} \dot{z} \stackrel{P}{=} (\dot{x} \dot{+} \dot{z})'$;
L03:	MP' \triangleright L02 \triangleright L01 \gg	$\dot{x}' \dot{+} \dot{z} \stackrel{P}{=} (\dot{x} \dot{+} \dot{z})'$;
L04:	Gen' \triangleright L03 \gg	$\dot{\forall}x: (\dot{x}' \dot{+} \dot{z} \stackrel{P}{=} (\dot{x} \dot{+} \dot{z})')$;
L05:	A4' @ $\dot{y} \gg$	$\dot{\forall}x: (\dot{x}' \dot{+} \dot{z} \stackrel{P}{=} (\dot{x} \dot{+} \dot{z})') \Rightarrow$ $(\dot{y}' \dot{+} \dot{z} \stackrel{P}{=} (\dot{y} \dot{+} \dot{z})')$;
L06:	MP' \triangleright L05 \triangleright L04 \gg	$\dot{y}' \dot{+} \dot{z} \stackrel{P}{=} (\dot{y} \dot{+} \dot{z})'$;

²⁰ [M Proposition 3.2(h)(i)] $\stackrel{\text{pyk}}{=} \text{"mendelson proposition three two h i"}]$

²¹ [M Proposition 3.2(h)_{(g)(ii)}] $\stackrel{\text{pyk}}{=} \text{"mendelson proposition three two h g ii"}]$

L07:	$\text{Gen}' \triangleright \text{L06} \gg$	$\dot{\forall}z: (\dot{y}' + \dot{z} \stackrel{\text{P}}{=} (\dot{y} + \dot{z})') \quad ;$
L08:	$\text{A4}' @ \dot{x} \gg$	$\dot{\forall}z: (\dot{y}' + \dot{z} \stackrel{\text{P}}{=} (\dot{y} + \dot{z})') \Rightarrow$ $\dot{y}' + \dot{x} \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad ;$
L09:	$\text{MP}' \triangleright \text{L08} \triangleright \text{L07} \gg$	$\dot{y}' + \dot{x} \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad \square$

We now have the necessary means to prove the induction step of M Proposition 3

6.9.3 M Proposition 3.2(h)(ii)

We then prove the induction step.

[M Proposition 3.2(h)(ii)]²²

[S' lemma M Proposition 3.2(h)(ii): $\dot{\forall}y: (\dot{x} + \dot{y} \stackrel{\text{P}}{=} \dot{y} + \dot{x} \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} \dot{y}' + \dot{x})$]

S' proof of M Proposition 3.2(h)(ii):

L01:	$\text{S6}' \gg$	$\dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{x} + \dot{y})' \quad ;$
L02:	M Proposition 3.2(g) \gg	$\dot{\forall}y: (\dot{x}' + \dot{y} \stackrel{\text{P}}{=} (\dot{x} + \dot{y})') \quad ;$
L03:	M Proposition 3.2(h) _{(g)(ii)} \triangleright L02 \gg	$\dot{y}' + \dot{x} \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad ;$
L04:	$\text{S2}' \gg$	$\dot{x} + \dot{y} \stackrel{\text{P}}{=} \dot{y} + \dot{x} \Rightarrow (\dot{x} + \dot{y})' \stackrel{\text{P}}{=}$ $(\dot{y} + \dot{x})' \quad ;$
L05:	M Proposition 3.2(c) \gg	$\dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{x} + \dot{y})' \Rightarrow (\dot{x} + \dot{y})' \stackrel{\text{P}}{=}$ $(\dot{y} + \dot{x})' \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad ;$
L06:	M Tautology A \triangleright L05 \gg	$(\dot{x} + \dot{y})' \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=}$ $(\dot{x} + \dot{y})' \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad ;$
L07:	M Tautology B \triangleright L04 \triangleright L06 \gg	$\dot{x} + \dot{y} \stackrel{\text{P}}{=} \dot{y} + \dot{x} \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=}$ $(\dot{x} + \dot{y})' \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad ;$
L08:	M Tautology A \triangleright L07 \gg	$\dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{x} + \dot{y})' \Rightarrow \dot{x} + \dot{y} \stackrel{\text{P}}{=}$ $\dot{y} + \dot{x} \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \quad ;$
L09:	$\text{MP}' \triangleright \text{L08} \triangleright \text{L01} \gg$	$\dot{x} + \dot{y} \stackrel{\text{P}}{=} \dot{y} + \dot{x} \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=}$ $(\dot{y} + \dot{x})' \quad ;$
L10:	M Proposition 3.2(d) \gg	$\dot{x} + \dot{y}' \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \Rightarrow \dot{y}' + \dot{x} \stackrel{\text{P}}{=}$ $(\dot{y} + \dot{x})' \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} \dot{y}' + \dot{x} \quad ;$
L11:	M Tautology B \triangleright L09 \triangleright L10 \gg	$\dot{x} + \dot{y} \stackrel{\text{P}}{=} \dot{y} + \dot{x} \Rightarrow \dot{y}' + \dot{x} \stackrel{\text{P}}{=}$ $(\dot{y} + \dot{x})' \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} \dot{y}' + \dot{x} \quad ;$
L12:	M Tautology A \triangleright L11 \gg	$\dot{y}' + \dot{x} \stackrel{\text{P}}{=} (\dot{y} + \dot{x})' \Rightarrow \dot{x} + \dot{y} \stackrel{\text{P}}{=}$ $\dot{y} + \dot{x} \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=} \dot{y}' + \dot{x} \quad ;$
L13:	$\text{MP}' \triangleright \text{L12} \triangleright \text{L03} \gg$	$\dot{x} + \dot{y} \stackrel{\text{P}}{=} \dot{y} + \dot{x} \Rightarrow \dot{x} + \dot{y}' \stackrel{\text{P}}{=}$ $\dot{y}' + \dot{x} \quad ;$

²²[M Proposition 3.2(h)(ii)]^{pyk} “mendelson proposition three two h ii”]

$$\text{L14: } \text{Gen}' \triangleright \text{L13} \gg \quad \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x} \Rightarrow \dot{x} \dot{+} \dot{y}' \stackrel{\text{P}}{=} \dot{y}' \dot{+} \dot{x}) \quad \square$$

We have thus proved the inductive step and can proceed to our final objective.

6.9.4 M Proposition 3.2(h)

And finally, we can now prove the objective of this report, namely the commutative law of addition.

[M Proposition 3.2(h)]²³

[S' lemma M Proposition 3.2(h): $\dot{\forall} \dot{x}: \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x})$]

S' proof of M Proposition 3.2(h):

$$\begin{array}{ll} \text{L01: } \text{S9}' \gg & \dot{\forall} \dot{x}: (\dot{x} \dot{+} \dot{0} \stackrel{\text{P}}{=} \dot{0} \dot{+} \dot{x}) \Rightarrow \\ & ((\dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x} \Rightarrow \dot{x} \dot{+} \dot{y}' \stackrel{\text{P}}{=} \dot{y}' \dot{+} \dot{x})) \Rightarrow \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x})) \quad ; \\ \text{L02: } \text{M Proposition 3.2(h)(i)} \gg & \dot{\forall} \dot{x}: (\dot{x} \dot{+} \dot{0} \stackrel{\text{P}}{=} \dot{0} \dot{+} \dot{x}) \quad ; \\ \text{L03: } \text{MP}' \triangleright \text{L01} \triangleright \text{L02} \gg & \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x} \Rightarrow \dot{x} \dot{+} \dot{y}' \stackrel{\text{P}}{=} \dot{y}' \dot{+} \dot{x}) \Rightarrow \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x}) \quad ; \\ \text{L04: } \text{M Proposition 3.2(h)(ii)} \gg & \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x} \Rightarrow \dot{x} \dot{+} \dot{y}' \stackrel{\text{P}}{=} \dot{y}' \dot{+} \dot{x}) \quad ; \\ \text{L05: } \text{MP}' \triangleright \text{L03} \triangleright \text{L04} \gg & \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x}) \quad ; \\ \text{L06: } \text{Gen}' \triangleright \text{L05} \gg & \dot{\forall} \dot{x}: \dot{\forall} \dot{y}: (\dot{x} \dot{+} \dot{y} \stackrel{\text{P}}{=} \dot{y} \dot{+} \dot{x}) \quad \square \end{array}$$

We have thus proved M Proposition 3.2(h), that is, the commutative law of addition.

7 Conclusion

In this report we have stated and proved a number of propositions from [4], p. 156. More precisely we have proved Proposition 3.2(a)-(h), except Proposition 3.2(e). The proofs have been carried out with the ambition of keeping things as simple as possible. We have succeeded in doing so, proving all propositions by the help of the theorems given in [Theory S'] defined on the [peano](#) page, two tautologies, and one lemma needed in the proof of our final theorem. The final theorem stated our original assignment, namely the commutativity of addition. We can thus conclude that we have solved the assignment properly.

Furthermore, for the inexperienced user and logician we have given a short and adequate introduction to the Logiweb system, axiomatic theories, and peano

²³[M Proposition 3.2(h)]^{pyk} “mendelson proposition three two h”]

arithmetic. The introduction should make the report self-contained, though, we have given references to relevant literature where needed.

A The Name of the Page

This defines the name of the page:

[peano commutativity $\stackrel{\text{pyk}}{=} \text{“peano commutativity”}$]

B Priority Table

Priority table

Preassociative

[peano commutativity], [base], [bracket * end bracket], [big bracket * end bracket], [math * end math], [**flush left** *], [x], [y], [z], [\bowtie *], [$\xrightarrow{*}$ *], [pyk], [tex], [name], [prio], [*, [T], [if(*, *, *)], [$\xRightarrow{*}$ *], [val], [claim], [\perp], [f(*)], [(*)^T], [F], [0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [a], [b], [c], [d], [e], [f], [g], [h], [i], [j], [k], [l], [m], [n], [o], [p], [q], [r], [s], [t], [u], [v], [w], [(*)^M], [If(*, *, *)], [array{*} * end array], [l], [c], [r], [empty], [($\ast := \ast$)], [$\mathcal{M}(\ast)$], [$\tilde{\mathcal{U}}(\ast)$], [$\mathcal{U}(\ast)$], [$\mathcal{U}^M(\ast)$], [**apply**(*, *)], [**apply**₁(*, *)], [identifier(*)], [identifier₁(*, *)], [array-plus(*, *)], [array-remove(*, *, *)], [array-put(*, *, *, *)], [array-add(*, *, *, *, *)], [bit(*, *)], [bit₁(*, *)], [rack], ["vector"], ["bibliography"], ["dictionary"], ["body"], ["codex"], ["expansion"], ["code"], ["cache"], ["diagnose"], ["pyk"], ["tex"], ["texname"], ["value"], ["message"], ["macro"], ["definition"], ["unpack"], ["claim"], ["priority"], ["lambda"], ["apply"], ["true"], ["if"], ["quote"], ["proclaim"], ["define"], ["introduce"], ["hide"], ["pre"], ["post"], [$\mathcal{E}(\ast, \ast, \ast)$], [$\mathcal{E}_2(\ast, \ast, \ast, \ast, \ast)$], [$\mathcal{E}_3(\ast, \ast, \ast, \ast, \ast)$], [$\mathcal{E}_4(\ast, \ast, \ast, \ast, \ast)$], [**lookup**(*, *, *)], [**abstract**(*, *, *, *)], [$\llbracket \ast \rrbracket$], [$\mathcal{M}(\ast, \ast, \ast)$], [$\mathcal{M}_2(\ast, \ast, \ast, \ast, \ast)$], [$\mathcal{M}^\ast(\ast, \ast, \ast)$], [macro], [s₀], [**zip**(*, *)], [**assoc**₁(*, *, *, *)], [(*)^P], [self], [($\ast \doteq \ast$)], [($\ast \dot{=}$ *)], [($\ast \dot{=} \ast$)], [($\ast \stackrel{\text{pyk}}{=}$ *)], [($\ast \stackrel{\text{tex}}{=}$ *)], [($\ast \stackrel{\text{name}}{=}$ *)], [**Priority table** *], [$\tilde{\mathcal{M}}_1$], [$\tilde{\mathcal{M}}_2(\ast)$], [$\tilde{\mathcal{M}}_3(\ast)$], [$\tilde{\mathcal{M}}_4(\ast, \ast, \ast, \ast, \ast)$], [$\tilde{\mathcal{M}}(\ast, \ast, \ast)$], [$\hat{\mathcal{Q}}(\ast, \ast, \ast)$], [$\hat{\mathcal{Q}}_2(\ast, \ast, \ast)$], [$\hat{\mathcal{Q}}_3(\ast, \ast, \ast, \ast, \ast)$], [$\hat{\mathcal{Q}}^\ast(\ast, \ast, \ast)$], [(*)], [**aspect**(*, *)], [**aspect**(*, *, *, *)], [(\ast)], [**tuple**₁(*)], [**tuple**₂(*)], [let₂(*, *)], [let₁(*, *)], [($\ast \stackrel{\text{claim}}{=}$ *)], [checker], [**check**(*, *)], [**check**₂(*, *, *)], [**check**₃(*, *, *)], [**check**^{*}(*, *)], [**check**₂^{*}(*, *, *)], [($\ast \cdot$)], [($\ast \cdot$)⁻], [($\ast \cdot$)^o], [msg], [($\ast \stackrel{\text{msg}}{=}$ *)], [<stmt>], [stmt], [($\ast \stackrel{\text{stmt}}{=}$ *)], [HeadNil'], [HeadPair'], [Transitivity'], [\perp], [Contra'], [T_E'], [L₁], [\boxtimes], [\mathcal{A}], [\mathcal{B}], [\mathcal{C}], [\mathcal{D}], [\mathcal{E}], [\mathcal{F}], [\mathcal{G}], [\mathcal{H}], [\mathcal{I}], [\mathcal{J}], [\mathcal{K}], [\mathcal{L}], [\mathcal{M}], [\mathcal{N}], [\mathcal{O}], [\mathcal{P}], [\mathcal{Q}], [\mathcal{R}], [\mathcal{S}], [\mathcal{T}], [\mathcal{U}], [\mathcal{V}], [\mathcal{W}], [\mathcal{X}], [\mathcal{Y}], [\mathcal{Z}], [($\ast := \ast$)], [($\ast \ast := \ast$)], [\emptyset], [Remainder], [(\ast)^v], [intro(*, *, *, *)], [intro(*, *, *)], [error(*, *)], [error₂(*, *)], [proof(*, *, *)], [proof₂(*, *)], [$\mathcal{S}(\ast, \ast)$], [$\mathcal{S}^I(\ast, \ast)$], [$\mathcal{S}^\triangleright(\ast, \ast)$], [$\mathcal{S}_1^\triangleright(\ast, \ast, \ast)$], [$\mathcal{S}^E(\ast, \ast)$], [$\mathcal{S}_1^E(\ast, \ast, \ast)$], [$\mathcal{S}^+(\ast, \ast)$], [$\mathcal{S}_1^+(\ast, \ast, \ast)$], [$\mathcal{S}^-(\ast, \ast)$], [$\mathcal{S}_1^-(\ast, \ast, \ast)$], [$\mathcal{S}^\ast(\ast, \ast)$], [$\mathcal{S}_1^\ast(\ast, \ast, \ast)$], [$\mathcal{S}_2^\ast(\ast, \ast, \ast, \ast)$], [$\mathcal{S}^\textcircled{\ast}(\ast, \ast)$], [$\mathcal{S}_1^\textcircled{\ast}(\ast, \ast, \ast)$], [$\mathcal{S}^+(\ast, \ast)$], [$\mathcal{S}_1^+(\ast, \ast, \ast, \ast)$], [$\mathcal{S}^\text{H}(\ast, \ast)$], [$\mathcal{S}_1^\text{H}(\ast, \ast, \ast, \ast)$], [$\mathcal{S}^{\text{i.e.}}(\ast, \ast)$], [$\mathcal{S}_1^{\text{i.e.}}(\ast, \ast, \ast, \ast)$], [$\mathcal{S}_2^{\text{i.e.}}(\ast, \ast, \ast, \ast, \ast)$], [$\mathcal{S}^\vee(\ast, \ast)$], [$\mathcal{S}_1^\vee(\ast, \ast, \ast, \ast)$], [$\mathcal{S}^\text{!}(\ast, \ast)$], [$\mathcal{S}_1^\text{!}(\ast, \ast, \ast)$], [$\mathcal{S}_2^\text{!}(\ast, \ast, \ast, \ast)$], [$\mathcal{T}(\ast)$], [claims(*, *, *)], [claims₂(*, *, *)], [<proof>], [proof], [**Lemma** * : *], [**Proof of** * : *], [**lemma** * : *], [**antilemma** * : *], [**rule** * : *], [**antirule** * : *], [verifier], [$\mathcal{V}_1(\ast)$], [$\mathcal{V}_2(\ast, \ast)$], [$\mathcal{V}_3(\ast, \ast, \ast, \ast)$], [$\mathcal{V}_4(\ast, \ast)$], [$\mathcal{V}_5(\ast, \ast, \ast, \ast)$], [$\mathcal{V}_6(\ast, \ast, \ast, \ast)$], [$\mathcal{V}_7(\ast, \ast, \ast, \ast)$], [Cut(*, *)], [Head_⊕(*)], [Tail_⊕(*)], [rule₁(*, *)], [rule(*, *)],

$[* \cdot *], [* \cdot_0 *], [* \dot{\cdot} *];$

Preassociative

$[* + *], [* +_0 *], [* +_1 *], [* - *], [* -_0 *], [* -_1 *], [* \dot{+} *];$

Preassociative

$[* \cup \{*\}], [* \cup *], [* \setminus \{*\}];$

Postassociative

$[* \dot{:} *], [* \dot{:} *], [* \ddot{:} *], [* \underline{+2*}], [* :: *], [* +2* *];$

Postassociative

$[*, *];$

Preassociative

$[* \overset{B}{\approx} *], [* \overset{D}{\approx} *], [* \overset{C}{\approx} *], [* \overset{P}{\approx} *], [* \approx *], [* = *], [* \overset{+}{\rightarrow} *], [* \overset{t}{=} *], [* \overset{t^*}{=} *], [* \overset{r}{=} *],$
 $[* \in_t *], [* \subseteq_T *], [* \overset{T}{=} *], [* \overset{s}{=} *], [* \text{free in } *], [* \text{free in}^* *], [* \text{free for } * \text{ in } *],$
 $[* \text{free for}^* * \text{ in } *], [* \in_c *], [* < *], [* <' *], [* \leq' *], [* \overset{P}{=} *], [* \mathcal{P}];$

Preassociative

$[\neg *], [\dot{\neg} *];$

Preassociative

$[* \wedge *], [* \ddot{\wedge} *], [* \tilde{\wedge} *], [* \wedge_c *], [* \dot{\wedge} *];$

Preassociative

$[* \vee *], [* \parallel *], [* \ddot{\vee} *], [* \dot{\vee} *];$

Preassociative

$[\dot{\forall}*: *], [\dot{\exists}*: *];$

Postassociative

$[* \rightrightarrows *], [* \Rightarrow *], [* \Leftrightarrow *];$

Postassociative

$[* : *], [* !*];$

Preassociative

$[* \left\{ \begin{array}{c} * \\ * \end{array} \right\}];$

Preassociative

$[\lambda * . *], [\Lambda *], [\text{if } * \text{ then } * \text{ else } *], [\text{let } * = * \text{ in } *], [\text{let } * \doteq * \text{ in } *];$

Preassociative

$[*^I], [*^\triangleright], [*^V], [*^+], [*^-], [*^*];$

Preassociative

$[* @ *], [* \triangleright *], [* \blacktriangleright *], [* \gg *];$

Postassociative

$[* \vdash *], [* \Vdash *], [* \text{i.e. } *];$

Preassociative

$[\forall*: *];$

Postassociative

$[* \oplus *];$

Postassociative

$[*, *];$

Preassociative

$[* \text{ proves } *];$

Preassociative

```

[* proof of * : *], [Line * : * >> *; *], [Last line * >> * □],
[Line * : Premise >> *; *], [Line * : Side-condition >> *; *], [Arbitrary >> *; *],
[Local >> * = *; *];
Postassociative
[* then *], [* [ * ]*];
Preassociative
[*&*];
Preassociative
[*\*]; End table

```

C Bibliography

- [1] Klaus Grue. *Introduction to Logiweb*. <http://www.diku.dk/~grue/logiweb/20050502/home/grue/check/GRD-2005-05-26-UTC-06-49-32-964854/>, 2005.
- [2] Klaus Grue. *A Logiweb base page*. <http://www.diku.dk/~grue/logiweb/20050502/home/grue/base/latest/>, 2005.
- [3] Klaus Grue. *Peano arithmetic*. <http://www.diku.dk/~grue/logiweb/20050502/home/grue/peano-axioms/latest/>, 2005.
- [4] E. Mendelson. *Introduction to Mathematical Logic*. Wadsworth and Brooks, 3. edition, 1987.